ICC

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

PROFINET IO Driver Manual

October 30, 2014

TABLE OF CONTENTS

1 PROFINET IO

1.1 Overview

The PROFINET IO device driver allows a controller to interact with the database via cyclic data exchange and acyclic read/write requests. The cyclic data exchange method is a high speed operation that bypasses the TCP/IP layer.

Other notes of interest are:

- Supports real time (RT) communication.
- 11 standard I/O modules are available for selection by the controller. The standard I/O modules follow the naming conventions of "IN:XXXX, OUT:YYYY", where XXXX is the number of input bytes to be sent to the controller, and YYYY is the number of output bytes to be sent to the driver.
- When using standard I/O modules, up to 512 bytes can be sent to the driver, and up to 512 bytes can be retrieved from the driver.
- To improve efficiency, the GSDML file can be modified by the user to support almost any combination of up to 1400 input bytes and up to 1400 output bytes. Refer to the PROFINET controller user's manual for the maximum supported input and output data lengths.
- The cyclic data connection sizes are limited by the driver only in that they must lie entirely within the database boundaries.
- No explicit driver module selection is required: the appropriate module will be selected automatically according to the controller's configuration.
- Eight separates slots (1…8) can be simultaneously utilized. The size of the I/O data is the sum of the modules in the occupied slots.
- The lowest recommended I/O cycle update time is 8ms. Lower update times are possible but will vary depending on the device's overall processing load.
- If a connection timeout occurs, the driver will trigger a timeout event. The timeout time is dictated by the PROFINET controller and is at a minimum, three times the IO cycle update time. The timeout value is also known as the "IO Cycle Watchdog" time.

1.2 Device Settings

<u>Device Name</u>

The device name / station name must be unique across the entire PROFINET network, because it is used by controllers to uniquely identify PROFINET devices. This string must conform to the device name requirements contained in the PROFINET specification.

1.3 Cyclic Data Access

When the PROFINET IO driver is added to the configuration, eight cyclic data slots are automatically included. Each slot allows for independent configuration of the produced and consumed database start address mapping. Note that while the produced and consumed data start addresses are set via the Configuration Studio, the sizes of these exchanged data packets are determined by the controller upon initial connection establishment.

Only those slots that will actually be accessed by the controller must be properly configured: unused slots may contain “dummy” database start addresses. Caution must be used when assigning addresses: depending on the sizes of the modules selected during hardware configuration, it is allowable for produced and consumed data to overlap both within slots and across slots (and the Configuration Studio therefore accommodates this configuration).

Produced Data Start Address

Defines the starting address in the database from which this slot’s produced data will be taken for transmission to the controller.

Consumed Data Start Address

Defines the starting address in the database to which this slot’s consumed data will be inserted after reception from the controller.

1.4 Acyclic Data Access

Any database value can be accessed via PROFINET acyclic services. To accomplish this, set the API to 0, Slot to 1, and SubSlot to 1. The record number/index value is equivalent to the desired database address, and the length is specified according to the number of bytes to access.

1.5 Customizing the GSDML

The GSDML xml file can be customized to fit the exact needs of the user. The only required item is a text editor (such as WordPad) and the latest GSDML file, available for download from http://www.iccdesigns.com. The following example demonstrates how to modify the number of input and output bytes for an existing module using a text editor and the Step 7 Hardware Configuration.

1.5.1 GSDML Modification Example

This section demonstrates how to modify an existing module to consume 123 (input) bytes and produce 45 (output) bytes.

1) Download the latest GSDML file from http://www.iccdesigns.com.
2) Open the GSDML file with a text editor such as WordPad.
3) Choose any module to edit. In this example, we’ll modify module “VM_1”.
4) Modify the ModuleIdentNumber from “0x00020002” to “0x01230045”. Notice that the first half of the ModuleIdentNumber represents the number of input bytes (0123) and the second half represents the number of output bytes (0045).
5) Locate the last reference of IN2_OUT2 and change the “Value” to “IN: 0123, OUT: 0045”.

6) Save the GSDML file.

7) Open the (Step 7) Hardware Configuration.

8) Navigate to *Options* → *Install GSD File…* and click Browse.

9) Select the folder containing the customized GSDML file.

10) Navigate to *Options* → *Update Catalog*.

11) Verify that the new module (“IN: 0123, OUT: 0045”) is now listed in the Virtual IO. If not, it may be necessary to close and reopen the Hardware Configuration and repeat step 10.

12) The new module is now ready to be inserted into Slot 1.

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

1600 Aspen Commons, Suite 210
Middleton, WI USA 53562-4720
Tel: [608] 831-1255   Fax: [608] 831-2045

http://www.iccdesigns.com

Printed in U.S.A